View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 83 (2016) 313 – 320

Procedia Computer Science

The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)

Reducing Energy Consumption for Distributed EM-based Clustering in Wireless Sensor Networks

John Tsiligaridis[a],Cesar Flores[b]

[a,b]*Heritage University,340 Fort Road ,Toppenish, WA 98948, USA.*

Abstract

Minimizing the energy consumption of wireless sensors is critical, yet a challenge for the design of wireless sensor networks (WSN). Energy is consumed in WSNs by sensing, communicating and processing. In various WSN applications, it is likely that communications are the major source of power consumption, rather than computation. Therefore, assuming that local processing is much less expensive than communicating, we focus on minimizing the number of transmissions for a distributed clustering problem in a sensor network. In our setup, each node in the network senses an environment that can be described as a mixture of Gaussians and each Gaussian component corresponds to one of the elementary conditions. For estimating the Gaussian components, we propose a distributed EM algorithm based on stochastic approximation (DEM-SA), and we study a trade-off between local processing and communication for the distributed clustering problem. DEM-SA reduces the traffic and contention in a WSN by keeping computations and communications local and avoiding the need for cycles through the network. Simulation results will be presented.

© 2016 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Conference Program Chairs

Keywords: Wireless Sensors; Distributed Systems; EM algorithm; Clustering

1. Introduction

Sensor networks are formed from a collection of sensing nodes which communicate with one another, typically through wireless channels, in order to collect spatially distributed data about their environment. Such networks have the potential to provide better quality data than single or small numbers of individual sensors in applications such as

* Corresponding author. Tel.: +509-865-8551; fax: +509-865-8679.
E-mail address: tsiligaridis_j@heritage.edu

1877-0509 © 2016 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.131

natural and built environmental monitoring, process monitoring, security and surveillance[1].

Sensor networks have a broad range of environmental sensing applications, including environment monitoring, vehicle tracking, collaborative processing of information, gathering data from spatially distributed sources, etc.[2].

The benefits of WSN include more efficient utilization of resources, better understanding of the behaviour of humans, natural and engineering systems, and increased safety and security.

Sensor networks have recently emerged as an important computing platform. Sensor Nodes are usually: (a) Typically less mobile and more densely deployed than mobile ad-hoc networks (MANETs). (b) Limited in processing, memory, and communication capabilities (c) Constrained in battery lifetime, (d) Left unattended e.g., in hostile environments, which makes it difficult impossible to re-charge or replace their batteries. Energy consumption in a sensor node can be due to either "useful" or "wasteful" sources. Useful energy consumption can be due to transmitting/ receiving data, processing query requests, and forwarding queries/data to neighboring nodes. Wasteful energy consumption can be due to idle listening to the media, retransmitting due to packet collisions, overhearing, and generating/handling control packets.

The sensors massively distributed small devices are equipped with sensing and processing capabilities that give us a new eye with which to explore our universe. Density estimation and unsupervised clustering are central first steps in exploratory data analysis[3,4,5]. The problem is what are the basic patterns and structures in the measured data. This problem has been addressed via maximum likelihood estimation for the centralized case. For the centralized case the data are stored and processed at a central location. On the contrary, here the data are assumed to be distributed across a collection of networked devices. We also suppose that the cost of computation at each node is much less that the cost of communication between nodes, implying that centralized data processing is very expensive and even infeasible for some applications due to high data requirements for communication, physical spatial constraints of the network, etc. In[11], a distributed EM algorithm based on a consensus approach is proposed. Although the idea is to approximate the class parameters in a distributed way, the parameterization is different from our paper, which leads to a different distributed algorithm and corresponding convergence properties. The main difference is that at each node, we observe a batch of data drawn from a normal mixture, while [11] considers only a single data point per node drawn from a normal component density that is in the mixture. Our algorithm is based on first using local information to compute local statistics using a batch of data at each node, and then using a stochastic approximation method to estimate global statistics. The class parameters (e.g., means and covariances of densities) are then obtained locally by combining the local statistics with the estimated global statistics at each node.

The distributed nature of the sensor network problem places emphasis on the tradeoff of local processing and communication. It is assumed that local processing is significantly less expensive than communication. The purpose of the presented distributed EM algorithm is to reduce the processing time by distributing computations amongst various nodes, minimizing data communication requirements by not transmitting the data themselves (often of high bandwidth), but by simply communicating sufficient statistics of very low bandwidth. These techniques have the potential to significantly improve energy efficiency and increase the lifetime of the network. Experiments are presented with two nodes with heterogeneous class proportions to illustrate some important benefits of this distributed approach that are not achievable with the centralized approach.

The paper is organized as follows: In section 1 there is an introduction to Statistics and Gaussian Mixture Models. Section 2 contains the standard EM algorithm for mixture density estimation. The distributed EM algorithm is included in section 3. Section 4 contains simulation experiments for the two nodes. Results are included in section 5.

2. Statistic, Gaussian Mixture Model

The Gaussian Mixture Models (GMM) are among the most statistically methods intensively used for density estimation and also for clustering. The GMM are considered as a simple linear superposition of Gaussian components, aimed at providing a richer class of density models than the single Gaussian. It is assumed that the individual data points are generated from a set of multivariate Gaussians. A GMM is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) Algorithm or the Maximum A posteriori (MAP) estimation from a well-trained prior model[6].

The Gaussian mixture model for the data is given by:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathrm{N}(x \mid \mu_k, \Sigma_k) \tag{1}$$

where each Gaussian density $N(x \mid \mu_k, \Sigma_k)$ is the component of the mixture and has the μ_κ and covariance Σ_k. Generally mixture models can be considered as linear combinations of other distributions. The π_κ are the mixing coefficients. The marginal density is given by $p(x) = \sum_{k=1}^{K} p(k)\, p(x \mid k)$ where $p(k)$ is the prior probability, and the density $p(x \mid k)$ is the conditional probability. So the mixing coefficients can be interpreted as prior probabilities. The labels of the mixtures are latent variables in this model. The posteriors probabilities (or responsibilities) for a given value of data can be computed as:

$$\gamma_k(x) = p(k \mid x) = \pi_k \mathrm{N}(x \mid \mu_k, \Sigma_k) / \sum_{j=1}^{K} \pi_j \mathrm{N}(x \mid \mu_j, \Sigma_j) \tag{2}$$

The maximum likelihood estimator maximizes the data log-likelihood given by:

$$\ln p(X \mid \pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln\Big(\sum_{k=1}^{K} \pi_k N(x_n \mid \mu_k, \Sigma_k)\Big) \tag{3}$$

It is well known that there is no closed form solution for the maximum likelihood.

3. EM Algorithm

The objective of the EM algorithm is to maximize the likelihood p(X|θ) of the data X drawn from an unknown distribution, given the model parameterized by θ:

$$\theta^* = \arg\max_{\theta} p(X \mid \theta) = \arg\max_{\theta} \prod_{n=1}^{N} p(x_n \mid \theta)$$

EM is an iterative procedure that is very sensitive to initial conditions! Therefore, there is a need for a good and fast initialization procedure.

The basic idea of EM is to :

A) Introduce a hidden variable such that its knowledge would simplify the $\max_{\theta} p(X \mid \theta)$

B) At each iteration of the algorithm:

E-Step: estimate the distribution of the hidden variable given the data and the current value of the parameters
M-Step: modify the parameters in order to maximize the joint distribution of the data and the hidden variable
In the E-step, the probability for each point to belong in each Gaussian is calculated. The M-step modifies the

parameters according to the posterior distribution of the hidden variables to maximize the data likelihood. Here, the optimization variables are the mixing coefficients, the means and covariances. Analytically the steps are:

(a) Initialize the m_k (means), Σ_k (covariances) and π_k (mixing coefficients)

(b) **E-step**: evaluate the posteriors as in (2) using the current parameter values

(c) **M-step**: re-estimate the parameters using the current posterior probabilities.

$$\mu_k^{new} = \sum_{n=1}^{N} \gamma(z_{nk}) x_n$$

$$\Sigma_k^{new} = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T \quad (4)$$

$$\pi_k^{new} = N_k / N$$

In (4), the class posterior probabilities are denoted as $\gamma(z_{nk})$.

(d) Compute the log likelihood as in[3]. The algorithm terminates according to a stopping criterion. A criterion often used in practice is convergence of the parameters or the log-likelihood. This algorithm monotonically improves the likelihood[7].

4. Distributed EM Algorithm

For the distributed computation, each node communicates with its neighbors and works independently. In the standard EM algorithm, the posterior probabilities are computed and the new parameter values (π_k, μ_k, Σ_k) are re-estimated using the current posterior probabilities in an iterative manner. We remark that all the data are used in each step of the computations. In the distributed setup, each node has access to its own collected data and does not have access to the data at other nodes. The distributed EM algorithm aims to estimate the means and the covariances of the clusters by performing the EM algorithm computations in a distributed way.
Let $\{y_n(m)\}_{n=1}^{N_m}$ denote the local data available at node m. We define the local sufficient statistics at node m as:

$$a_k^t(m) = \frac{1}{N_m} \sum_{n=1}^{N_m} a_{n,k}^t(m)$$

$$b_k^t(m) = \sum_{n=1}^{N_m} a_{n,k}^t(m) y_n(m)$$

$$C_k^t(m) = \sum_{n=1}^{N_m} a_{n,k}^t(m)(y_n(m) - \hat{\mu}_k^t(m))(y_n(m) - \hat{\mu}_k^t(m))^T \text{ (LOCAL)}$$

And define the global sufficient statistics as:

$$a_k^t = \sum_{m=1}^{M} N_m a_k^t(m), b_k^t = \sum_{m=1}^{M} b_k^t(m), C_k^t = \sum_{m=1}^{M} C_k^t(m)$$

With these definitions in place, the parameter estimates can be calculated as:

$$\mu_k^{t+1} = \frac{b_k^{t+1}}{a_k^{t+1}}$$

$$\Sigma_k^{t+1} = \frac{C_k^{t+1}}{a_k^{t+1}}$$

While the local sufficient statistics can be locally computed at each node, the global sufficient statistics cannot. The distributed EM (DEM) algorithm proposed in[8] cycles through the network and perform incremental E and M steps for each node using only the local data at each node and the global sufficient statistics are estimated by using the local sufficient statistics passed from the previous node in the cycle. The DEM algorithm of[8] proceeds in a cyclic fashion by passing messages between nodes in the order $(1,2,...,M,1,2,...,M,etc.)$. The processing and communication is carried out at each node in succession. The disadvantage of this distributed algorithm is that the global sufficient statistics need to be transmitted through the entire network. When the network has many nodes and/or the complexity increases, the delay incurred by the distributed cyclic nature of the algorithm causes slow convergence of the algorithms. Furthermore, it is unclear if the particular network topology can be leveraged effectively to speed up the algorithm.

An alternative method that has gained significant popularity in distributed algorithms that uses the underlying network topology is to perform nearest-neighbor information exchanging. In this manner, local computations can approximate global computations by passing information around the network in a computationally effective manner. In addition, wireless sensor networks are inherently amenable to these types of computations because wireless communications is more reliable locally. In the sense of distributed computation, each node only communicates with its neighbors and works independently. In the standard EM algorithm, it can be found that the local summary quantities can be calculated locally, while the global summary quantities cannot be calculated locally. However, the global summary quantities can be viewed as the averages of the local summary quantities from all nodes. This idea has connections to control theory [9,10], in which stability of these types of distributed algorithms can be guaranteed under mild assumptions. In the E-step of this algorithm, each sensor node independently calculates local sufficient statistics by using local observations. An information exchange with nearest neighbors is used to diffuse local sufficient statistics and estimate global sufficient statistics in each node. Through this distributed processing, each node gradually diffuses local information over the entire network and the estimate of global sufficient statistics is asymptotically obtained. In the M-step, each sensor node uses the estimated global sufficient statistics to update the parameters of the Gaussian mixtures, which can maximize the log-likelihood in the same way as in the standard EM algorithm. Because the nearest-neighbor information processing only requires that each node communicate with its neighbors, this distributed EM algorithm is scalable and robust. Each node m can compute the local sufficient statistics $(a_k^t(m), b_k^t(m), C_k^t(m))$. Each node m also maintains an estimate of the global sufficient statistics (a_k^t, b_k^t, C_k^t) denoted as $(\hat{a}_k^t(m), \hat{b}_k^t(m), \hat{C}_k^t(m))$. The information exchange step for node m takes as input the local sufficient statistics and the neighbors' estimated global sufficient statistics, and outputs the updated estimated global sufficient statistics, which are also sent to neighbors for the M-step. The estimated global sufficient statistics are used to calculate the parameters at each node m, i.e.,

$$\hat{\mu}_k^{t+1}(m) = \frac{\hat{b}_k^t(m)}{\hat{a}_k^t(m)} \quad \text{(ESTIMATE)}$$

$$\hat{\Sigma}_k^{t+1}(m) = \frac{\hat{C}_k^t(m)}{\hat{a}_k^t(m)}$$

The information exchange formula is based on a stochastic approximation procedure that considers the nearest-neighbors of node m, $N(m)$, and is given as:

$$\begin{aligned}
\hat{a}_k^{t+1}(m) &= \hat{a}_k^t(m) + \varepsilon\left(\sum_{l\in N(m)}(\hat{a}_k^t(l) - \hat{a}_k^t(m)) + (N_m a_k^t(m) - \hat{a}_k^t(m))\right) \\
\hat{b}_k^{t+1}(m) &= \hat{b}_k^t(m) + \varepsilon\left(\sum_{l\in N(m)}(\hat{b}_k^t(l) - \hat{b}_k^t(m)) + (b_k^t(m) - \hat{b}_k^t(m))\right) \\
\hat{C}_k^{t+1}(m) &= \hat{C}_k^t(m) + \varepsilon\left(\sum_{l\in N(m)}(\hat{C}_k^t(l) - \hat{C}_k^t(m)) + (C_k^t(m) - \hat{C}_k^t(m))\right)
\end{aligned} \tag{5}$$

where $\varepsilon \leq \frac{1}{d_{\max}}$ is the updating rate, and $d_{\max}$ is the maximum node degree in the graph.

This distributed EM algorithm with stochastic approximation updates (DEM-SA) works as follows:

Initialize parameter θ^0, t=0

REPEAT

E-step: Each node calculates the local sufficient statistics $(a_k^t(m), b_k^t(m), C_k^t(m))$ as in (LOCAL).

Stochastic Approximation Step: Each node calculates the estimates of the global sufficient statistics $(\hat{a}_k^t(m), \hat{b}_k^t(m), \hat{C}_k^t(m))$ as in (5)

M-step: Each node calculates the parameters θ^t using (ESTIMATE).

t = t+1

UNTIL CONVERGENCE

5. Simulation

The simulation is based on DEM-SA with two nodes. Two-dimensional data are generated from the multivariate normal distributions and assigned to two clusters, for each node. Each node/sensor has different class proportions.

Figure 1 shows the imbalanced distribution of data in sensor 1. The cluster on the left has 5 data points and the cluster on the right has 995 data points

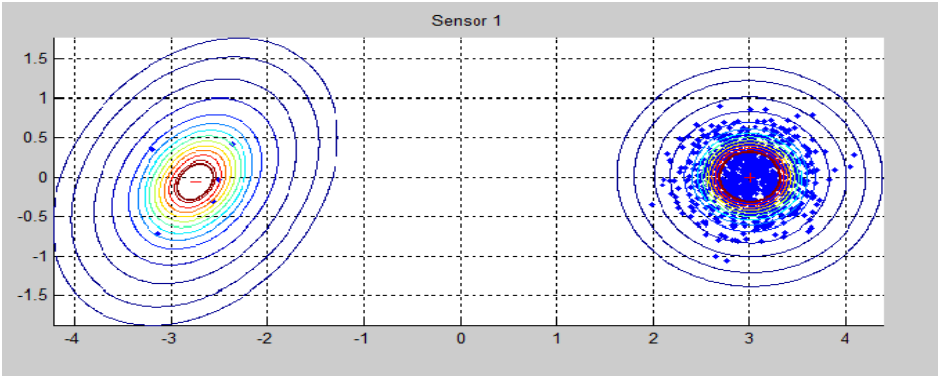

Figure 1: Data distribution in sensor 1

Figure 2 shows the imbalanced distribution of data in sensor 2. The cluster on the left has 995 data points and the cluster on the right has only 5 data points. Thus, although the model parameters are the same for both sensors, the class proportions are flipped.

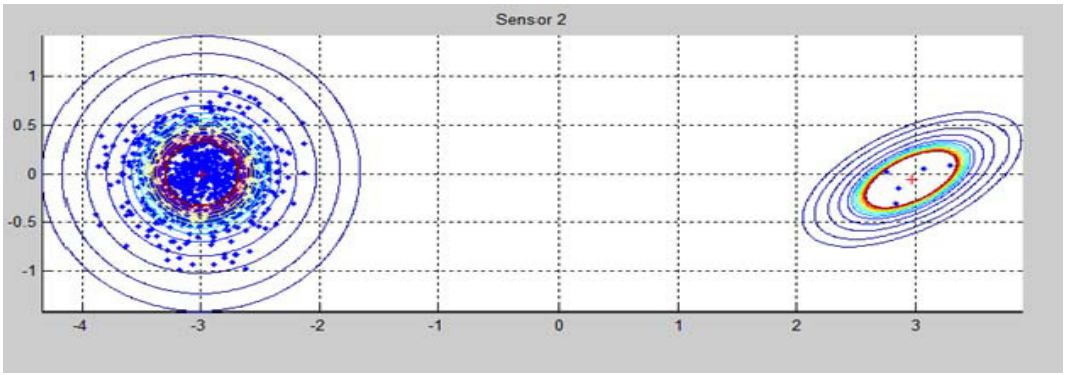

Figure 2: Data distribution in sensor 2

For two nodes, various experiments have been done to observe the performance of DEM-SA. The first experiment was to study the estimation error in the means for both nodes, with and without information sharing. The results are shown in Figure 3. From this, we observe that information sharing allows faster convergence than the case without information sharing.

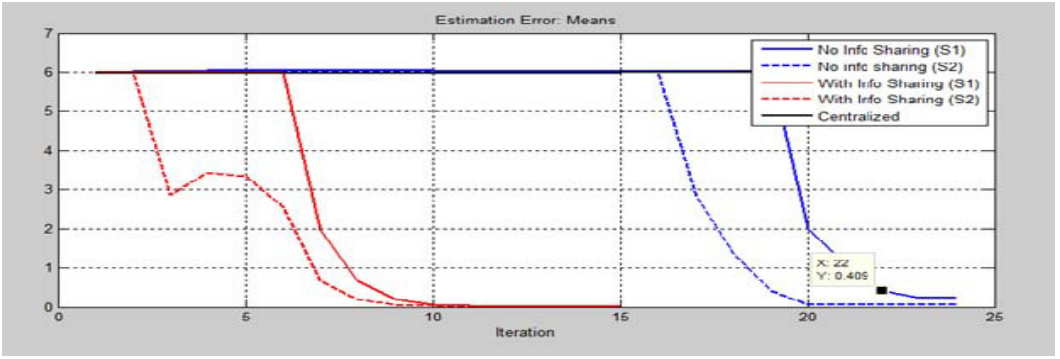

Figure 3: Estimation error of means for the two sensors

Finally, the mean errors are observed again with each system's sensors' results averaged. Once again, we see that information sharing results in better performance than without information sharing; these results are shown in Figure 4. Note that the performance of the centralized EM algorithm is not improving as a function of iteration because it gets stuck to a local maximum.

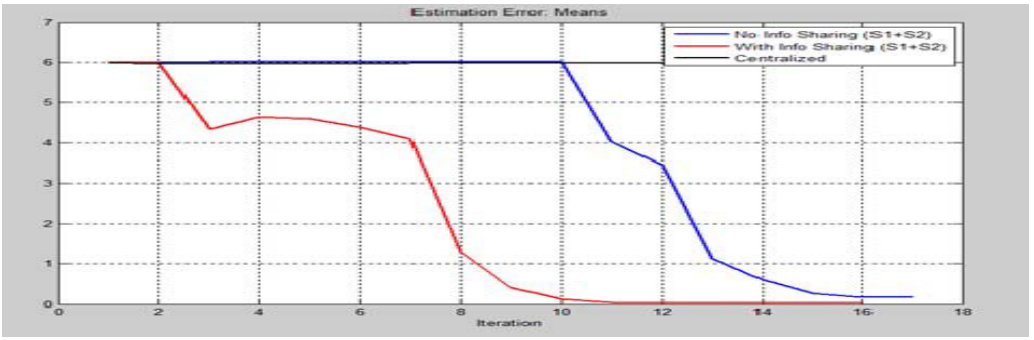

Figure 4: Overall error in means for systems with and without information sharing**.**

6. Conclusion

Two types of solutions are considered in order to reduce energy consumption. The first is offered by several MAC protocols (for the wasteful sources). The second is offered from a number of protocols for the useful sources. The hierarchical clustering techniques can aid in reducing useful energy consumption. Clustering can also reduce the communication overhead.

This paper is based on distributed EM algorithm with nearest-neighbour stochastic approximations. DEM-SA reduces the traffic and contention in a WSN by keeping computations and communications local and avoiding the need for cycles through the network. Future work will involve experiments on more complex heterogeneous networks.

References

1. I. Stojmenovic. *Handbook of Sensor Networks: Algorithms and Architectures*, John Wiley & Sons Inc, Hoboken, New Jersey, USA, 2005
2. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Next century challenges: Scalable coordination in sensor networks,” in *Proc. ACM/IEEE Int. Conf. Mobile Comput. Netw.*, Seattle, WA, 1999, pp.263–270
3. R. D. Nowak, “Distributed EM algorithms for density estimation and clustering in sensor networks,” *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2245–2253, Aug. 2003
4. Y. Sheng, X. Hu, and P. Ramanathan, “Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor networks,” in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw.*, Los Angeles, CA, Apr. 2005, pp. 181–188.
5. R. Kamimura, “Cooperative information maximization with Gaussian activation functions for self-organizing maps,” *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 909–918, Jul. 2006.
6. D. Reynolds *Gaussian Mixture Models*, MIT Lincoln Lab, Lexington, MA,USA
7. A. Dempster,N. Laird, and D. Rubin, “Maximum likelihood estimation from incomplete data via the EM algorithm,” *J. Roy. Statist. Soc.*, vol.39, pp. 1–38, 1977
8. R. Nowak, “Distributed EM algorithms for Density Estimation and Clustering in Sensor Networks”, *IEEE Trans. Signal Processing*, vol. 51, no. 8, pp. 2245-2253, Aug. 2003
9. N. A. Lynch, *Distributed Algorithms*. San Francisco, CA: Morgan Kaufmann, 1996
10. R. Olfati-Saber, and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delay,” *IEEE Trans. Automatic Control*, vol.49, no.9, pp.101–115, Sep. 2004
11. S. Pereira, S. Barabarossa, and A. Pages-Zamora, “Consensus for distributed EM based clustering in WSNs” ,6th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM’ 10),pp. 45-48,Oct. 2010